APS: Active Pretraining with Successor Features

Hao Liu [1] **Pieter Abbeel** [1]

Abstract

We introduce a new unsupervised pretraining objective for reinforcement learning. During the unsupervised reward-free pretraining phase, the agent maximizes mutual information between tasks and states induced by the policy. Our key contribution is a novel lower bound of this intractable quantity. We show that by reinterpreting and combining variational successor features (Hansen et al., 2020) with nonparametric entropy maximization (Liu & Abbeel, 2021), the intractable mutual information can be efficiently optimized. The proposed method Active Pretraining with Successor Feature (APS) explores the environment via nonparametric entropy maximization, and the explored data can be efficiently leveraged to learn behavior by variational successor features. APS addresses the limitations of existing mutual information maximization based and entropy maximization based unsupervised RL, and combines the best of both worlds. When evaluated on the Atari 100k data-efficiency benchmark, our approach significantly outperforms previous methods combining unsupervised pretraining with task-specific finetuning.

1. Introduction

Deep unsupervised pretraining has achieved remarkable success in various frontier AI domains from natural language processing (Devlin et al., 2019; Peters et al., 2018; Brown et al., 2020) to computer vision (He et al., 2020; Chen et al., 2020a). The pre-trained models can quickly solve downstream tasks through few-shot fine-tuning (Brown et al., 2020; Chen et al., 2020b).

In reinforcement learning (RL), however, training from scratch to maximize extrinsic reward is still the dominant paradigm. Despite RL having made significant progress in playing video games (Mnih et al., 2015; Schrittwieser et al., 2019; Vinyals et al., 2019; Badia et al., 2020a) and solving complex robotic control tasks (Andrychowicz et al., 2017; Akkaya et al., 2019), RL algorithms have to be trained from scratch to maximize extrinsic return for every encountered task. This is in sharp contrast with how intelligent creatures quickly adapt to new tasks by leveraging previously acquired behaviors.

In order to bridge this gap, unsupervised pretraining RL has gained interest recently, from state-based (Gregor et al., 2016; Eysenbach et al., 2019; Sharma et al., 2020; Mutti et al., 2020) to pixel-based RL (Hansen et al., 2020; Liu & Abbeel, 2021; Campos et al., 2021). In unsupervised pretraining RL, the agent is allowed to train for a long period without access to environment reward, and then got exposed to reward during testing. The goal of pretraining is to have data efficient adaptation for some downstream task defined in the form of rewards.

State-of-the-art unsupervised RL methods consider various ways of designing the so called intrinsic reward (Barto et al., 2004; Barto, 2013; Gregor et al., 2016; Achiam & Sastry, 2017), with the goal that maximizing this intrinsic return can encourage meaningful behavior in the absence of external rewards. There are two lines of work in this direction, we will discuss their advantages and limitations, and show that a novel combination yields an effective algorithm which brings the best of both world.

The first category is based on maximizing the mutual information between task variables ($p(z)$) and their behavior in terms of state visitation ($p(s)$) to encourage learning distinguishable task conditioned behaviors, which has been shown effective in state-based RL (Gregor et al., 2016; Eysenbach et al., 2019) and visual RL (Hansen et al., 2020). VISR proposed in Hansen et al. (2020) is the prior state-of-the-art in this category. The objective of VISR is $\max I(s; z) = \max H(z) - H(s|z)$ where z is sampled from a fixed distribution. VISR proposes a successor features based variational approximation to maximize a variational lower bound of the intractable conditional entropy $-H(s|z)$. The advantage of VISR is that its successor features can quickly adapt to new tasks. Despite its effectiveness, the fundamental problem faced by VISR is lack of exploration.

Another category is based on maximizing the entropy of

[1]University of California, Berkeley, CA, USA. Correspondence to: Hao Liu <hao.liu@cs.berkeley.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

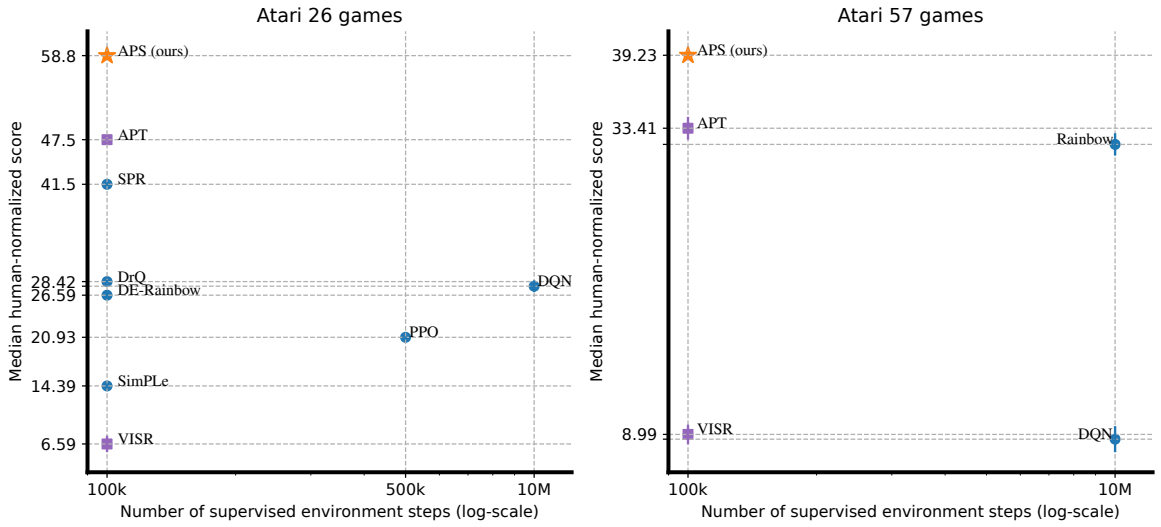

Figure 1: Median of human normalized score on the 26 Atari games considered by Kaiser et al. (2020) (left) and the Atari 57 games considered in Mnih et al. (2015)(right). Fully supervised RL baselines are shown in circle. RL with unsupervised pretraining are shown in square. APS significantly outperforms all of the fully supervised and unsupervised pre-trained RL methods. Baselines: Rainbow (Hessel et al., 2018), SimPLe (Kaiser et al., 2020), APT (Liu & Abbeel, 2021), Data-efficient Rainbow (Kielak, 2020), DrQ (Kostrikov et al., 2020), VISR (Hansen et al., 2020), CURL (Laskin et al., 2020), and SPR (Schwarzer et al., 2021).

the states induced by the policy $\max H(s)$. Maximizing state entropy has been shown to work well in state-based domains (Hazan et al., 2019; Mutti et al., 2020) and pixel-based domains (Liu & Abbeel, 2021). It is also shown to be provably efficient under certain assumptions (Hazan et al., 2019). The prior state-of-the-art APT by Liu & Abbeel (2021) show maximizing a particle-based entropy in a lower dimensional abstraction space can boost data efficiency and asymptotic performance. However, the issues with APT are that it is purely exploratory and task-agnostic and lacks of the notion of task variables, making it more difficult to adapt to new tasks compared with task-conditioning policies.

Our main contribution is to address the issues of APT and VISR by combining them together in a novel way. To do so, we consider the alternative direction of maximizing mutual information between states and task variables $I(s; z) = H(s) - H(s|z)$, the state entropy $H(s)$ encourages exploration while the conditional entropy encourages the agent to learn task conditioned behaviors. Prior work that considered this objective had to either make the strong assumption that the distribution over states can be approximated with the stationary state-distribution of the policy (Sharma et al., 2020) or rely on the challenging density modeling to derive a tractable lower bound (Sharma et al., 2020; Campos et al., 2020). We show that the intractable conditional entropy, $-H(s|z)$ can be lower bounded and optimized by learning successor features. We use APT to maximize the state entropy $H(s)$ in an abstract representation space. Building upon this insight, we propose Active Pretraining with Successor Features (APS) since the agent is encouraged to actively explore and leverage the experience to learn behavior. By doing so, we experimentally find that they address the limitations of each other and significantly improve each other.

We evaluate our approach on the Atari benchmark (Bellemare et al., 2013) where we apply APS to DrQ (Kostrikov et al., 2020) and test its performance after fine-tuning for 100K supervised environment steps. The results are shown in Figure 1. On the 26 Atari games considered by (Kaiser et al., 2020), our fine-tuning significantly boosts the data-efficiency of DrQ, achieving 106% relative improvement. On the full suite of Atari 57 games (Mnih et al., 2015), fine-tuning APS pre-trained models significantly outperforms prior state-of-the-art, achieving human median score $3\times$ higher than DQN trained with 10M supervised environment steps and outperforms previous methods combining unsupervised pretraining with task-specific finetuning.

2. Related Work

Our work falls under the category of mutual information maximization for unsupervised behavior learning. Unsupervised discovering of a set of task-agnostic behaviors by means of seeking to maximize an extrinsic reward has been explored in the the evolutionary computation community (Lehman & Stanley, 2011a;b). This has long been studied as intrinsic motivation (Barto, 2013; Barto et al., 2004), often with the goal of encouraging exploration (Simsek & Barto, 2006; Oudeyer & Kaplan, 2009). Entropy maximization in state space has been used to encourage

Table 1: Comparing methods for pretraining RL in no reward setting. VISR (Hansen et al., 2020), APT (Liu & Abbeel, 2021), MEPOL (Mutti et al., 2020), DIYAN (Eysenbach et al., 2019), DADS (Sharma et al., 2020), EDL (Campos et al., 2020). Exploration: the model can explore efficiently. Off-policy: the model is off-policy RL. Visual: the method works well in visual RL, e.g., Atari games. Task: the model conditions on latent task variables z. * means only in state-based RL.

Algorithm	Objective	Exploration	Visual	Task	Off-policy	Pre-Trained Model
APT	$\max \mathrm{H}(s)$	✓	✓	✗	✓	$\pi(a\|s), Q(s,a)$
VISR	$\max \mathrm{H}(z) - \mathrm{H}(z\|s)$	✗	✓	✓	✓	$\psi(s,z)$, $\phi(s)$
MEPOL	$\max \mathrm{H}(s)$	✓*	✗	✗	✗	$\pi(a\|s)$
DIAYN	$\max -\mathrm{H}(z\|s) + \mathrm{H}(a\|z,s)$	✗	✗	✓	✗	$\pi(a\|s,z)$
EDL	$\max \mathrm{H}(s) - \mathrm{H}(s\|z)$	✓*	✗	✓	✓	$\pi(a\|s,z), q(s'\|s,z)$
DADS	$\max \mathrm{H}(s) - \mathrm{H}(s\|z)$	✓	✗	✓	✗	$\pi(a\|s,z), q(s'\|s,z)$
APS	$\max \mathrm{H}(s) - \mathrm{H}(s\|z)$	✓	✓	✓	✓	$\psi(s,z)$, $\phi(s)$

$\psi(s)$: successor features, $\phi(s)$: state feature ($i.e.$, the representation of states).

exploration in state RL (Hazan et al., 2019; Mutti et al., 2020; Seo et al., 2021) and visual RL (Liu & Abbeel, 2021; Yarats et al., 2021). Maximizing the mutual information between latent variable policies and their behavior in terms of state visitation has been used as an objective for discovering meaningful behaviors (Houthooft et al., 2016a; Mohamed & Rezende, 2015; Gregor et al., 2016; Houthooft et al., 2016b; Eysenbach et al., 2019; Warde-Farley et al., 2019). Sharma et al. (2020) consider a similar decomposition of mutual information, namely, $I(s;z) = H(s) - H(z|s)$, however, they assume $p(s|z) \approx p(s)$ to derive a different lower-bound of the marginal entropy. Different from Sharma et al. (2020), Campos et al. (2020) propose to first maximize $H(s)$ via maximum entropy estimation (Hazan et al., 2019; Lee et al., 2019) then learn behaviors, this method relies on a density model that provides an estimate of how many times an action has been taken in similar states. These methods are also only shown to work from explicit state-representations, and it is nonobvious how to modify them to work from pixels. The work by Badia et al. (2020b) also considers k-nearest neighbor based count bonus to encourage exploration, yielding improved performance on Atari games. This heuristically defined count-based bonus has been shown to be an effective unsupervised pretraining objective for RL (Campos et al., 2021). Machado et al. (2020) show the norm of learned successor features can be used to incentivize exploration as a reward bonus. Our work differs in that we jointly maximize the entropy and learn successor features.

3. Preliminaries

Reinforcement learning considers the problem of finding an optimal policy for an agent that interacts with an uncertain environment and collects reward per action. The goal of the agent is to maximize its cumulative reward.

Formally, this problem can be viewed as a Markov decision process (MDP) defined by $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, r, \gamma)$ where $\mathcal{S} \subseteq \mathbb{R}^{n_s}$ is a set of n_s-dimensional states, $\mathcal{A} \subseteq \mathbb{R}^{n_a}$ is a set of n_a-dimensional actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the state transition probability distribution. $\rho_0 : \mathcal{S} \to [0,1]$ is the distribution over initial states, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0,1)$ is the discount factor. At environment states $s \in \mathcal{S}$, the agent take actions $a \in \mathcal{A}$, in the (unknown) environment dynamics defined by the transition probability $T(s'|s,a)$, and the reward function yields a reward immediately following the action a_t performed in state s_t. We define the discounted return $G(s_t, a_t) = \sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l})$ as the discounted sum of future rewards collected by the agent. In value-based reinforcement learning, the agent learns learns an estimate of the expected discounted return, a.k.a, state-action value function.

$$Q^\pi(s,a) = \mathbb{E}_{\substack{s_t=s \\ a_t=a}} \left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}, s_{t+l+1}) \right].$$

3.1. Successor Features

Successor features (Dayan, 1993; Kulkarni et al., 2016; Barreto et al., 2017; 2018) assume that there exist features $\phi(s,a,s') \in R^d$ such that the reward function which specifies a task of interest can be written as

$$r(s,a,s') = \phi(s,a,s')^T w,$$

where $w \in R^d$ is the task vector that specify how desirable each feature component is.

The key observation is that the state-action value function can be decomposed as a linear form (Barreto et al., 2017)

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_{\substack{s_t=s \\ a_t=a}} \left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi(s_{i+1}, a_{i+1}, s'_{i+1}) \right]^T w \\ &\equiv \psi^\pi(s,a)^T w, \end{aligned}$$

where $\psi^\pi(s,a)$ are the successor features of π. Intuitively, $\psi(s,a)$ can be seen as a generalization of $Q(s,a)$ to multi-dimensional value function with reward $\phi(s,a,s')$

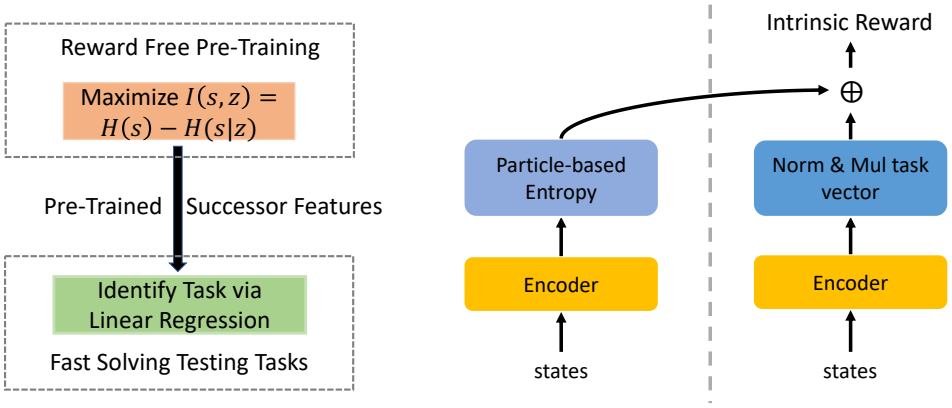

Figure 2: Diagram of the proposed method APS. On the left shows the concept of APS, during reward-free pretraining phase, reinforcement learning is deployed to maximize the mutual information between the states induced by policy and the task variables. During testing, the pre-trained state features can identify the downstream task by solving a linear regression problem , the pre-trained task conditioning successor features can then quickly adapt to and solve the task. On the right shows the components of APS. APS consists of maximizing state entropy in an abstract representation space (exploration, $\max H(s)$) and leveraging explored data to learn task conditioning behaviors (exploitation, $\max -H(s|z)$).

4. Method

We first introduce two techniques which our method builds upon in Section 4.1 and Section 4.2 and discuss their limitations. We provide preliminary evidence of the limitations in Section 4.3. Then we propose APS in Section 4.4 to address their limitations.

4.1. Variational Intrinsic Successor Features (VISR)

The variational intrinsic successor features (VISR) maximizes the mutual information(I) between some policy-conditioning variable (z) and the states induced by the conditioned policy,

$$I(z;s) = H(z) - H(z|s),$$

where it is common to assume z is drawn from a fixed distribution for the purposes of training stability (Eysenbach et al., 2019; Hansen et al., 2020).

This simplifies the objective to minimizing the conditional entropy of the conditioning variable, where s is sampled uniformly over the trajectories induced by π_θ.

$$\sum_{z,s} p(s,z) \log p(z|s) = \mathbb{E}_{s,z}[\log p(z|s)],$$

A variational lower bound is proposed to address the intractable objective,

$$J_{\text{VISR}}(\theta) = -\mathbb{E}_{s,z}[\log q(z|s)],$$

where $q(z|s)$ is a variational approximation. REINFORCE algorithm is used to learn the policy parameters by treating $\log q(z|s)$ as intrinsic reward. The variational parameters can be optimized by maximizing log likelihood of samples.

The key observation made by Hansen et al. (2020) is restricting conditioning vectors z to correspond to task-vectors w of the successor features formulation $z \equiv w$. To satisfy this requirement, one can restrict the task vectors w and features $\phi(s)$ to be unit length and paremeterizing the discriminator $q(z|s)$ as the Von Mises-Fisher distribution with a scale parameter of 1.

$$r_{\text{VISR}}(s,a,s') = \log q(w|s) = \phi(s)^T w.$$

VISR has the rapid task inference mechanism provided by successor features with the ability of mutual information maximization methods to learn many diverse behaviors in an unsupervised way. Despite its effectiveness as demonstrated in Hansen et al. (2020), VISR suffers from inefficient exploration. This issue limits the further applications of VISR in challenging tasks.

4.2. Unsupervised Active Pretraining (APT)

The objective of unsupervised active pretraining (APT) is to maximize the entropy of the states induced by the policy, which is computed in a lower dimensional abstract representation space.

$$J_{\text{APT}}(\theta) = H(h) = \sum_s p(h) \log p(h), \quad h = f(s),$$

where $f : R^{n_s} \to R^{n_h}$ is a mapping that maps observations s to lower dimensional representations h. In their work, Liu & Abbeel (2021) learns the encoder by contrastive representation learning.

With the learned representation, APT shows the entropy of h can be approximated by a particle-based entropy estimation (Singh et al., 2003; Beirlant, 1997), which is based on

the distance between each particle $h_i = f(s_i)$ and its k-th nearest neighbor $h_i^\star$.

$$H(h) \approx H_{\text{APT}}(h) \propto \sum_{i=1}^{n} \log \|h_i - h_i^\star\|_{n_z}^{n_z}.$$

This estimator is asymptotically unbiased and consistent $\lim_{n\to\infty} H_{\text{APT}}(s) = H(s)$.

It helps stabilizing training and improving convergence in practice to average over all k nearest neighbors (Liu & Abbeel, 2021).

$$\hat{H}_{\text{APT}}(h) = \sum_{i=1}^{n} \log \left(1 + \frac{1}{k} \sum_{h_i^j \in \text{N}_k(h_i)} \|h_i - h_i^j\|_{n_h}^{n_h} \right),$$

where $\text{N}_k(\cdot)$ denotes the k nearest neighbors.

For a batch of transitions $\{(s, a, s')\}$ sampled from the replay buffer, each abstract representation $f(s')$ is treated as a particle and we associate each transition with a intrinsic reward given by

$$r_{\text{APT}}(s, a, s') = \log \left(1 + \frac{1}{k} \sum_{h^{(j)} \in \text{N}_k(h)} \|h - h^{(j)}\|_{n_z}^{n_z} \right)$$
$$\text{where } h = f_\theta(s'). \tag{1}$$

While APT achieves prior state-of-the-art performance in DeepMind control suite and Atari games, it does not conditions on latent variables (e.g. task) to capture important task information during pretraining, making it inefficient to quickly identity downstream task when exposed to task specific reward function.

4.3. Empirical Evidence of the Limitations of Existing Models

In this section we present two multi-step grid-world environments to illustrate the drawbacks of APT and VISR, and highlight the importance of both exploration and task inference. The environments, implemented with the pycolab game engine (Stepleton, 2017), are depicted shown in Figure 3, and are fully observable to the agent. At each episode, the agent starts from a randomly initialized location in the top left corner, with the task of navigating to the target location shown in orange. To do so, the agent has to first pick up a key(green, purple area) that opens the closed passageway. The easy task shown in left of Figure 3 has one key and one corresponding passageway while the hard task has two key-passageway pairs. We evaluate the agent in terms of success rates. During evaluation, the agent receives an intermediate reward 1 for picking up key and 10 for completing the task. The hierarchical task presents a challenge to algorithms using only exploration bonus or successor features, as the exploratory policy is unlikely to quickly adapt to the task specific reward and the successor features is likely to never explore the space sufficiently.

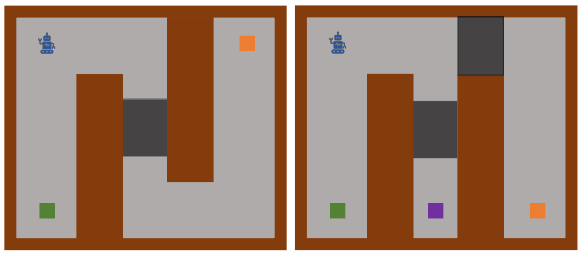

Figure 3: The passageway gridworld environments used in our experiments. On the left, the agent needs to fetch the key first by navigating to the green location to unlock the closed passageway (shown in black). Similarly, on the right, there is an additional key-passageway pair. The agent must fetch the key (shown in purple) to unlock the upper right passageway.

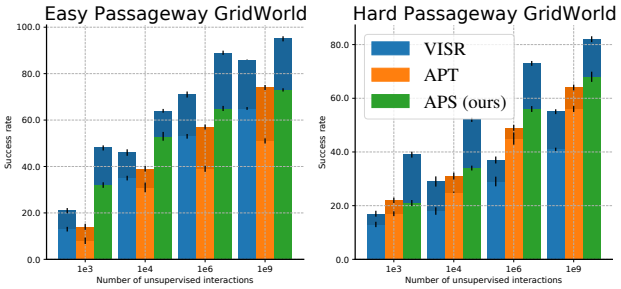

Figure 4: Performance of different methods on the gridworld environments in Figure 3. The results are recorded during testing phase after pretraining for a number of unsupervised interactions. The success rate are aggregated over 10 random seeds. The bottom of each bar is the zero-shot testing performance while the top is the fine-tuned performance.

Figure 4 shows the success rate of each method. APT performs worse than VISR at the easy level, possibly because successor features can quickly adapt to the downstream reward. On the other hand, APT significantly outperforms VISR at the hard level which requires a exploratory policy. Despite the simplicity, these two gridworld environments already highlight the weakness of each method. This observation confirms that existing formulations either fail due to inefficient exploration or slow adaption, and motivates our study of alternative methods for behavior discovery.

4.4. Active Pre-training with Successor Features

To address the issues of APT and VISR, we consider maximizing the mutual information between task variable (z) drawn from a fixed distribution and the states induced by the conditioned policy.

$$I(z; s) = H(s) - H(s|z).$$

Algorithm 1: Training APS

Randomly Initialize ϕ network // $L2$ normalized output
Randomly Initialize ψ network // $dim(output) = \#A \times dim(W)$
for $e := 1, \infty$ **do**
  sample w from $L2$ normalized $\mathcal{N}(0, I(dim(W)))$ // uniform ball
  $Q(\cdot, a|w) \leftarrow \psi(\cdot, a, w)^\top w, \forall a \in A$
  for $t := 1, T$ **do**
    Receive observation s_t from environment
    $a_t \leftarrow \epsilon$-greedy policy based on $Q(s_t, \cdot|w)$
    Take action a_t, receive observation s_{t+1} and reward ~~r_t~~ from environment
    $a' = \arg\max_a \psi(s_{t+1}, a, w)^\top w$
    Compute $r_{\text{APS}}(s_t, a, s_{t+1})$ with Equation (7) // intrinsic reward to $\max I(s; z)$
    $y = r_{\text{APS}}(s_t, a, s_{t+1}) + \gamma \psi(s_{t+1}, a', w)^\top w$
    $loss_\psi = (\psi(s_t, a_t, w)^\top w - y_i)^2$
    $loss_\phi = -\phi(s_t)^\top w$ // minimize Von-Mises NLL
    Gradient descent step on ψ and ϕ // minibatch in practice
  end
end

The intuition is that the $H(s)$ encourages the agent to explore novel states while $H(s|z)$ encourages the agent to leverage the collected data to capture task information.

Directly optimizing $H(s)$ is intractable because the true distribution of state is unknown, as introduced in Section 4.2, APT (Liu & Abbeel, 2021) is an effective approach for maximizing $H(s)$ in high-dimensional state space. We use APT to perform entropy maximization.

$$r_{\text{APS}}^{\text{exploration}}(s, a, s') = \log\left(1 + \frac{1}{k} \sum_{h^{(j)} \in \text{N}_k(h)} \|h - h^{(j)}\|_{n_h}^{n_h}\right) \quad \text{where } h = f_\theta(s'). \tag{2}$$

As introduced in Section 4.1, VISR (Hansen et al., 2020) is a variational based approach for maximizing $-H(z|s)$. However, maximizing $-H(z|s)$ is not directly applicable to our case where the goal is to maximize $-H(s|z)$.

This intractable conditional entropy can be lower-bounded by a variational approximation,

$$F = -H(s|z) \geq \mathbb{E}_{s,z}\left[\log q(s|z)\right].$$

This is because of the variational lower bound (Barber & Agakov, 2003).

$$\begin{aligned} F &= \sum_{s,z} p(s,z) \log p(s|z) \\ &= \sum_{s,z} p(s,z) \log p(s|z) + \sum_{s,z} p(s,z) \log q(s|z) \\ &\quad - \sum_{s,z} p(s,z) \log q(s|z) \\ &= \sum_{s,z} p(s,z) \log q(s|z) + \sum_{z} p(z) D_{\text{KL}}(p(\cdot|z)||q(\cdot|z)) \\ &\geq \sum_{s,z} p(s,z) \log q(s|z) \\ &= \mathbb{E}_{s,z}[\log q(s|z)] \end{aligned} \tag{3}$$

Our key observation is that Von Mises-Fisher distribution is symmetric to its parametrization, by restricting $z \equiv w$ similarly to VISR, the reward can be written as

$$r_{\text{APS}}^{\text{exploitation}}(s, a, s') = \log q(s|w) = \phi(s)^T w. \tag{4}$$

We find it helps training by sharing the weights between encoders f and ϕ. The encoder is trained by minimizing the negative log likelihood of Von-Mises distribution $q(s|w)$ over the data.

$$L = -\mathbb{E}_{s,w}\left[\log q(s|w)\right] = -\mathbb{E}_{s,w}\left[\phi(s_t)^\top w\right]. \tag{5}$$

Note that the proposed method is independent from the choices of representation learning for f, *e.g.*, one can use an inverse dynamic model (Pathak et al., 2017; Burda et al., 2019) to learn the neural encoder, which we leave for future work.

Put Equation (2) and Equation (4) together, our intrinsic reward can be written as

$$r_{\text{APS}}(s,a,s') = r_{\text{APS}}^{\text{exploitation}}(s,a,s') + r_{\text{APS}}^{\text{exploration}}(s,a,s') \quad (6)$$

$$= \phi(s)^T w + \log\left(1 + \frac{1}{k}\sum_{h^{(j)}\in \text{N}_k(h)} \|h - h^{(j)}\|_{n_h}^{n_h}\right)$$

$$\text{where } h = \phi(s'), \quad (7)$$

The output layer of ϕ is L2 normalized, task vector w is randomly sampled from a uniform distribution over the unit circle.

Table 1 positions our new approach with respect to existing ones. Figure 2 shows the resulting model. Training proceeds as in other algorithms maximizing mutual information: by randomly sampling a task vector w and then trying to infer the state produced by the conditioned policy from the task vector. Algorithm 1 shows the pseudo-code of APS, we highlight the changes from VISR to APS in color.

4.5. Implementation Details

We largely follow Hansen et al. (2020) for hyperparameters used in our Atari experiments, with the following three exceptions. We use the four layers convolutional network from Kostrikov et al. (2020) as the encoder ϕ and f. We change the output dimension of the encoder from 50 to 5 in order to match the dimension used in VISR. While VISR incorporated LSTM (Hochreiter & Schmidhuber, 1997) we excluded it for simplicity and accelerating research. We use ELU nonlinearities (Clevert et al., 2016) in between convolutional layers. We do not use the distributed training setup in Hansen et al. (2020), after every roll-out of 10 steps, the experiences are added to a replay buffer. This replay buffer is used to calculate all of the losses and change the weights of the network. The task vector w is also resampled every 10 steps. We use n-step Q-learning with $n = 10$.

Following Hansen et al. (2020), we condition successor features on task vector, making $\psi(s,a,w)$ a UVFA (Borsa et al., 2019; Schaul et al., 2015). We use the Adam optimizer (Kingma & Ba, 2015) with an learning rate 0.0001. We use discount factor $\gamma = .99$. Standard batch size of 32. ψ is coupled with a target network (Mnih et al., 2015), with an update period of 100 updates.

5. Results

We test APS on the full suite of 57 Atari games (Bellemare et al., 2013) and the sample-efficient Atari setting (Kaiser et al., 2020; van Hasselt et al., 2019) which consists of the 26 easiest games in the Atari suite (as judged by above random performance for their algorithm).

We follow the evaluation setting in VISR (Hansen et al., 2020) and APT (Liu & Abbeel, 2021), agents are allowed a long unsupervised training phase (250M steps) without access to rewards, followed by a short test phase with rewards. The test phase contains 100K environment steps – equivalent to 400k frames, or just under two hours – compared to the typical standard of 500M environment steps, or roughly 39 days of experience. We normalize the episodic return with respect to expert human scores to account for different scales of scores in each game, as done in previous works. The human-normalized performance of an agent on a game is calculated as $\frac{\text{agent score} - \text{random score}}{\text{human score} - \text{random score}}$ and aggregated across games by mean or median.

When testing the pre-trained successor features ψ, we need to find task vector w from the rewards. To do so, we rollout 10 episodes (or 40K steps, whichever comes first) with the trained APS, each conditioned on a task vector chosen uniformly on a 5-dimensional sphere. From these initial episodes, we combine the data across all episodes and solve the linear regression problem. Then we fine-tune the pre-trained model for 60K steps with the inferred task vector, and the average returns are compared.

A full list of scores and aggregate metrics on the Atari 26 subset is presented in Table 2. The results on the full 57 Atari games suite is presented in Supplementary Material. For consistency with previous works, we report human and random scores from (Hessel et al., 2018).

In the data-limited setting, APS achieves super-human performance on eight games and achieves scores higher than previous state-of-the-arts.

In the full suite setting, APS achieves super-human performance on 15 games, compared to a maximum of 12 for any previous methods and achieves scores significantly higher than any previous methods.

6. Analysis

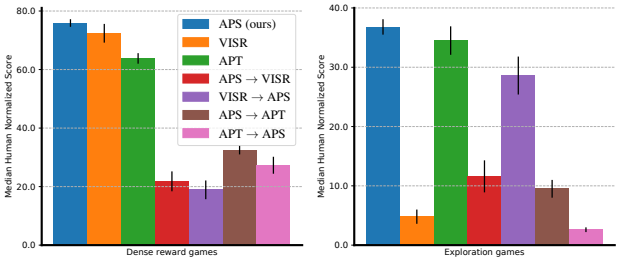

Figure 5: Scores of different methods and their variants on the 26 Atari games considered by Kaiser et al. (2020). $X \to Y$ denotes training method Y using the data collected by method X at the same time.

Table 2: Performance of different methods on the 26 Atari games considered by (Kaiser et al., 2020) after 100K environment steps. The results are recorded at the end of training and averaged over 5 random seeds for APS. APS outperforms prior methods on all aggregate metrics, and exceeds expert human performance on 8 out of 26 games while using a similar amount of experience.

Game	Random	Human	SimPLe	DER	CURL	DrQ	SPR	VISR	APT	APS (ours)
Alien	227.8	7127.7	616,9	739.9	558.2	771.2	801.5	364.4	**2614.8**	934.9
Amidar	5.8	1719.5	88.0	188.6	142.1	102.8	176.3	186.0	**211.5**	178.4
Assault	222.4	742.0	527.2	431.2	600.6	452.4	571.0	**12091.1**	891.5	413.3
Asterix	210.0	8503.3	1128.3	470.8	734.5	603.5	977.8	**6216.7**	185.5	1159.7
Bank Heist	14.2	753.1	34.2	51.0	131.6	168.9	380.9	71.3	**416.7**	262.7
BattleZone	2360.0	37187.5	5184.4	10124.6	14870.0	12954.0	16651.0	7072.7	7065.1	**26920.1**
Boxing	0.1	12.1	9.1	0.2	1.2	6.0	35.8	13.4	21.3	**36.3**
Breakout	1.7	30.5	16.4	1.9	4.9	16.1	17.1	17.9	10.9	**19.1**
ChopperCommand	811.0	7387.8	1246.9	861.8	1058.5	780.3	974.8	800.8	317.0	**2517.0**
Crazy Climber	10780.5	23829.4	62583.6	16185.2	12146.5	20516.5	42923.6	49373.9	44128.0	**67328.1**
Demon Attack	107805	35829.4	62583.6	16185.3	12146.5	20516.5	42923.6	**8994.9**	5071.8	7989.0
Freeway	0.0	29.6	20.3	27.9	26.7	9.8	24.4	-12.1	**29.9**	27.1
Frostbite	65.2	4334.7	254.7	866.8	1181.3	331.1	**1821.5**	230.9	1796.1	496.5
Gopher	257.6	2412.5	771.0	349.5	669.3	636.3	715.2	498.6	**2590.4**	2386.5
Hero	1027.0	30826.4	2656.6	6857.0	6279.3	3736.3	7019.2	663.5	6789.1	**12189.3**
Jamesbond	29.0	302.8	125.3	301.6	471.0	236.0	365.4	484.4	356.1	**622.3**
Kangaroo	52.0	3035.0	323.1	779.3	872.5	940.6	3276.4	1761.9	412.0	**5280.1**
Krull	1598.0	2665.5	**4539.9**	2851.5	4229.6	4018.1	2688.9	3142.5	2312.0	4496.0
Kung Fu Master	258.5	22736.3	17257.2	14346.1	14307.8	9111.0	13192.7	16754.9	17357.0	**22412.0**
Ms Pacman	307.3	6951.6	1480.0	1204.1	1465.5	960.5	1313.2	558.5	**2827.1**	2092.3
Pong	-20.7	14.6	**12.8**	-19.3	-16.5	-8.5	-5.9	-26.2	-8.0	12.5
Private Eye	24.9	69571.3	58.3	97.8	218.4	-13.6	**124.0**	98.3	96.1	117.9
Qbert	163.9	13455.0	1288.8	1152.9	1042.4	854.4	669.1	666.3	17671.2	**19271.4**
Road Runner	11.5	7845.0	5640.6	9600.0	5661.0	8895.1	**14220.5**	6146.7	4782.1	5919.0
Seaquest	68.4	42054.7	683.3	354.1	384.5	301.2	583.1	706.6	2116.7	**4209.7**
Up N Down	533.4	11693.2	3350.3	2877.4	2955.2	3180.8	**28138.5**	10037.6	8289.4	4911.9
Mean Human-Norm'd	0.000	1.000	44.3	28.5	38.1	35.7	70.4	64.31	69.55	**99.04**
Median Human-Norm'd	0.000	1.000	14.4	16.1	17.5	26.8	41.5	12.36	47.50	**58.80**
# Superhuman	0	N/A	2	2	2	2	7	6	7	**8**

Contribution of Exploration and Exploitation In order to measure the contributions of components in our method, we aim to answer the following two questions in this ablation study. Compared with APT ($\max H(s)$), is the improvement solely coming from better fast task solving induced by $\max -H(s|z)$ and the exploration is the same? Compared with VISR ($\max H(z) - H(z|s)$), is the improvement solely coming from better exploration due to $\max H(s) - H(s|z)$ and the task solving ability is the same?

We separate Atari 26 subset into two categories. Dense reward games in which exploration is simple and exploration games which require exploration. In addition to train the model as before, we simultaneously train another model using the same data, $e.g.$ APS $\rightarrow$ APT denotes when training APS simultaneously training APT using the same data as APS. As shown in Figure 5, on dense reward games, APS $\rightarrow$ APT performs better than APT $\rightarrow$ APS. On exploration games, APS $\rightarrow$ APT significantly outperforms APT $\rightarrow$ APS. Similarly APS $\rightarrow$ VISR performs better than the other way around. Together, the results indicate that entropy maximization and variational successor features improves each other in a nontrivial way, and both are important to the performance gain of APS.

Table 3: Scores on the 26 Atari games for variants of APS, VISR, and APT. Scores of considered variants are averaged over 3 random seeds.

Variant	Human-Normalized Score	
	mean	median
APS	**99.04**	**58.80**
APS w/o fine-tune	81.41	49.18
VISR (controlled, w/ fine-tune)	68.95	31.87
APT (controlled, w/o fine-tune)	58.23	19.85
APS w/o shared encoder	87.59	51.45

Fine-Tuning Helps Improve Performance We remove fine-tuning from APS that is we evaluate its zero-shot performance, the same as in Hansen et al. (2020). We also employ APS's fine-tuning scheme to VISR, namely 250M (without access to rewards, followed by a short task identify phase (40K steps) and a fine-tune phase (60K steps). The results shown in Table 3 demonstrate that fine-tuning can boost performance. APS w/o fine-tune outperforms all controlled baselines, including VISR w/ fine-tune.

Shared Encoder Can Boost Data-Efficiency We investigate the effect of using ϕ as the encoder f. To do so, we consider a variant of APS that learns the encoder f as in APT

by contrastive representation learning. The performance of this variant is denoted as APS w/o shared encoder shown in Table 3. Sharing encoder can boost data efficiency, we attribute the effectiveness to ϕ better captures the relevant information which is helpful for computing intrinsic reward. We leave the investigation of using other representation learning methods as future work.

7. Conclusion

In this paper, we propose a new unsupervised pretraining method for RL. It addresses the limitations of prior mutual information maximization-based and entropy maximization-based methods and combines the best of both worlds. Empirically, APS achieves state-of-the-art performance on the Atari benchmark, demonstrating significant improvements over prior work.

Our work demonstrates the benefit of leveraging state entropy maximization data for task-conditioned skill discovery. We are excited about the improved performance by decomposing mutual information as $H(s) - H(s|z)$ and optimizing them by particle-based entropy and variational successor features. In the future, it is worth studying how to combine approaches designed for maximizing the alternative direction $-H(z|s)$ with the particle-based entropy maximization.

8. Acknowledgment

We thank members of Berkeley Artificial Intelligence Research (BAIR) Lab for many insightful discussions. This work was supported by Berkeley Deep Drive, the Open Philanthropy Project, and Intel.

References

Achiam, J. and Sastry, S. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5048–5058, 2017.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 507–517. PMLR, 2020a.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., and Blundell, C. Never give up: Learning directed exploration strategies. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b.

Barber, D. and Agakov, F. V. The im algorithm: A variational approach to information maximization. In *Advances in neural information processing systems*, 2003.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., Silver, D., and van Hasselt, H. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4055–4065, 2017.

Barreto, A., Borsa, D., Quan, J., Schaul, T., Silver, D., Hessel, M., Mankowitz, D. J., Zídek, A., and Munos, R. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 510–519. PMLR, 2018.

Barto, A. G. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47. Springer, 2013.

Barto, A. G., Singh, S., and Chentanez, N. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, pp. 112–19. Piscataway, NJ, 2004.

Beirlant, J. Nonparametric entropy estimation: An overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39, 1997.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D. J., van Hasselt, H., Munos, R., Silver, D., and Schaul, T. Universal successor features approximators. In *7th International*

Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Burda, Y., Edwards, H., Pathak, D., Storkey, A. J., Darrell, T., and Efros, A. A. Large-scale study of curiosity-driven learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i-Nieto, X., and Torres, J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1317–1327. PMLR, 2020.

Campos, V., Sprechmann, P., Hansen, S. S., Barreto, A., Blundell, C., Vitvitskyi, A., Kapturowski, S., and Badia, A. P. Coverage as a principle for discovering transferable behavior in reinforcement learning, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b.

Clevert, D., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Hansen, S., Dabney, W., Barreto, A., Warde-Farley, D., de Wiele, T. V., and Mnih, V. Fast task inference with variational intrinsic successor features. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Hazan, E., Kakade, S. M., Singh, K., and Soest, A. V. Provably efficient maximum entropy exploration. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691. PMLR, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9726–9735. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00975.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3215–3222. AAAI Press, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Curiosity-driven exploration in deep reinforcement learning via bayesian neural networks. 2016a.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., Turck, F. D., and Abbeel, P. VIME: variational information maximizing exploration. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1109–1117, 2016b.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model based reinforcement learning for atari. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Kielak, K. Do recent advancements in model-based deep reinforcement learning really improve data efficiency? *arXiv preprint arXiv:2003.10181*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Kulkarni, T. D., Saeedi, A., Gautam, S., and Gershman, S. J. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016.

Laskin, M., Srinivas, A., and Abbeel, P. CURL: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 2020.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Lehman, J. and Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Lehman, J. and Stanley, K. O. Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*, pp. 37–56. Springer, 2011b.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. *arXiv preprint arXiv:2103.04551*, 2021.

Machado, M. C., Bellemare, M. G., and Bowling, M. Count-based exploration with the successor representation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5125–5133. AAAI Press, 2020.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2125–2133, 2015.

Mutti, M., Pratissoli, L., and Restelli, M. A policy gradient method for task-agnostic exploration. *arXiv preprint arXiv:2007.04640*, 2020.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1312–1320. JMLR.org, 2015.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., and Lee, K. State entropy maximization with random encoders for efficient exploration. *arXiv preprint arXiv:2102.09430*, 2021.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Simsek, Ö. and Barto, A. G. An intrinsic reward mechanism for efficient exploration. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pp. 833–840. ACM, 2006. doi: 10.1145/1143844.1143949.

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23 (3-4):301–321, 2003.

Stepleton, T. The pycolab game engine. `https://github.com/deepmind/pycolab`, 2017.

van Hasselt, H., Hessel, M., and Aslanides, J. When to use parametric models in reinforcement learning? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14322–14333, 2019.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Warde-Farley, D., de Wiele, T. V., Kulkarni, T. D., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021.